Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (currently amended): A data-processing device for processing in parallel a plurality of independent processes, comprising:

a program memory having stored therein at least one compiled program with a multiplicity N of independent processes, the compiled program including information on parallelism and including a multiplicity of bundles with a plurality of instructions of a process, the instructions of a bundle being executable in parallel;

a branching control unit connected to and addressing said program memory;

a register for storing flags and data which are switched in dependence on a process being executed;

a program flow control unit connected to said branching control unit, said program flow control unit controlling a fetching of bundles to be processed in parallel from said program memory, controlling said branching control unit, and

BEST AVAILABLE COPY

controlling an output of instructions to be processed in

parallel in dependence on information contained in the

instructions and included in a compiling time of the program;

a number N of instruction buffers being connected in parallel

downstream of said program memory for storing instructions

read out from said program memory, an instruction bundle being

read into one of said instruction buffers and a second

instruction bundle associated with a different process being

read into another one of said instruction buffers; and

an instruction output selector being connected to and

controlled by said ~~process~~ program flow control unit causing

said instruction output selector to read out instructions from

said instruction buffers and output N instructions in

parallel, said instruction output selector having a

multiplexer logic and selecting based on a determination by

said program flow control unit, in a first case, ~~one of either~~

one instruction from a first instruction buffer and one

instruction from a second instruction buffer for execution in

parallel and, in a second case, two instructions from one of

said first and second instruction buffers for execution in

parallel, said program flow control unit choosing between said

first case or said second case based on at least one of

priority information, NOP information and information of the

current process.

Claims 2 and 3 (canceled)

Claim 4 (original): The data-processing device according to

claim 1, which comprises N instruction decoders for decoding

the instructions being output.

Claim 5 (original): The data-processing device according to

claim 1, which comprises at least two instruction-execution

units for outputting the N decoded instructions.

Claim 6 (original): The data-processing device according to

claim 5, which comprises a data memory and at least two buses

connecting said N instruction-execution units to said data

memory.

Claim 7 (previously presented): The data-processing device

according to claim 1, which comprises a plurality of

instruction-execution units connected to said program flow

control unit and configured to execute the instructions of one

or more bundles in parallel.

Applic. No. 09/760,405
Response Dated September 18, 2006
Responsive to Office Action of May 17, 2006

Claim 8 (original): The data-processing device according to claim 1, wherein said branching control unit is configured to output an address pointer for addressing a bundle.

Claim 9 (original): The data-processing device according to claim 1, wherein the branching control unit comprises:

a first multiplexer and a second multiplexer;

an adder; and

N program counters; and

wherein said program flow control unit feeds a number of instructions in a bundle to said adder and said adder adds an address pointer and the number of instructions;

wherein said program flow control unit feeds addresses for program jumps or function calls and a process number to said first multiplexer;

said first multiplexer writing either the output signal of said adder or the addresses for program jumps or function calls into said program counter assigned to the active process; and

a content of said program counter assigned to the currently

active process is output as a new address pointer via said

second multiplexer which is controlled using the process

number supplied.

Claim 10 (original): The data-processing device according to

claim 1, wherein said program flow control unit is configured

to receive via a subbus of an output bus of said program

memory at least one of the following:

at least one bit for indicating the parallel execution of

instructions;

at least one bit for indicating the length of the following

instruction bundle;

the indication of one or more NOPs in the instruction bundles;

a priority of the processes of the instructions.

Claim 11 (previously presented): The data-processing device

according to claim 1, wherein a process is called with a run

instruction assigning a process number, a priority and a

memory address of a starting point of the process in the

program memory.

Claim 12 (original): The data-processing device according to

claim 1, wherein said data-processing device is a network

processor for processing layer 1 to 7 of protocol stacks in

applications including LAN, ATM switches, IP routers, and

frame relays based on a system selected from the group

consisting of DSL, Ethernet, and cable modems.

Claim 13 (new): A data-processing device for processing in

parallel a plurality of independent processes, comprising:

a program memory having stored therein at least one compiled

program with a multiplicity N of independent processes, the

compiled program including information on parallelism and

including a multiplicity of bundles with a plurality of

instructions of a process, the instructions of a bundle being

executable in parallel;

a branching control unit connected to and addressing said

program memory;

a register for storing flags and data which are switched in

dependence on a process being executed;

a program flow control unit connected to said branching

control unit, said program flow control unit controlling a

fetching of bundles to be processed in parallel from said

program memory, controlling said branching control unit, and

controlling an output of instructions to be processed in

parallel in dependence on information contained in the

instructions and included in a compiling time of the program;

a number N of instruction buffers being connected in parallel

downstream of said program memory for storing instructions

read out from said program memory, an instruction bundle being

read into one of said instruction buffers and a second

instruction bundle associated with a different process being

read into another one of said instruction buffers; and

an instruction output selector being connected to and

controlled by said program flow control unit causing said

instruction output selector to read out instructions from said

instruction buffers and output N instructions in parallel,

said instruction output selector having a multiplexer logic

and selecting based on a determination by said program flow

control unit, in a first case, one instruction from a first

instruction buffer and one instruction from a second

instruction buffer for execution in parallel and, in a second

case, two instructions from one of said first and second

instruction buffers for execution in parallel, said program

flow control unit choosing said first case in order to replace

an NOP instruction in one of said first instruction buffer or

said second instruction buffer by an instruction from the

other buffer.

Claim 14 (new): A data-processing device for processing in

parallel a plurality of independent processes, comprising:

a program memory having stored therein at least one compiled

program with a multiplicity N of independent processes, the

compiled program including information on parallelism and

including a multiplicity of bundles with a plurality of

instructions of a process, the instructions of a bundle being

executable in parallel;

a branching control unit connected to and addressing said

program memory;

a register for storing flags and data which are switched in

dependence on a process being executed;

a program flow control unit connected to said branching

control unit, said program flow control unit controlling a

fetching of bundles to be processed in parallel from said

program memory, controlling said branching control unit, and

controlling an output of instructions to be processed in parallel in dependence on information contained in the instructions and included in a compiling time of the program;

a number N of instruction buffers being connected in parallel downstream of said program memory for storing instructions read out from said program memory, an instruction bundle being read into one of said instruction buffers and a second instruction bundle associated with a different process being read into another one of said instruction buffers; and

an instruction output selector being connected to and controlled by said program flow control unit causing said instruction output selector to read out instructions from said instruction buffers and output N instructions in parallel, said instruction output selector having a multiplexer logic and selecting based on a determination by said program flow control unit, in a first case, one instruction from a first instruction buffer and one instruction from a second instruction buffer for execution in parallel and, in a second case, two instructions from one of said first and second instruction buffers for execution in parallel, said program flow control unit choosing between said first case and said second case, based on a priority assigned to each of the different processes.

Claim 15 (new): The data processing device of claim 1,
further comprising at least two instruction-execution units
for receiving the instructions selected by said instruction
output selector, each of said at least two instruction-
execution units being able to execute instructions output by
either of said first instruction buffer and said second
instruction buffer.

BEST AVAILABLE COPY